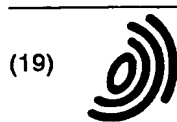


CITED BY APPLICANT



(19)

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 1 223 504 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
17.07.2002 Bulletin 2002/29

(51) Int Cl.7: **G06F 3/033**(21) Application number: **02250226.4**(22) Date of filing: **14.01.2002**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR**
Designated Extension States:
AL LT LV MK RO SI

(72) Inventor: **Watanabe, Yoshiaki**
Fuji Photo Film Co., Ltd.
Saitama (JP)

(74) Representative: **Stevens, Jason Paul**
Frank B. Dehn & Co.
179 Queen Victoria Street
London EC4V 4EL (GB)

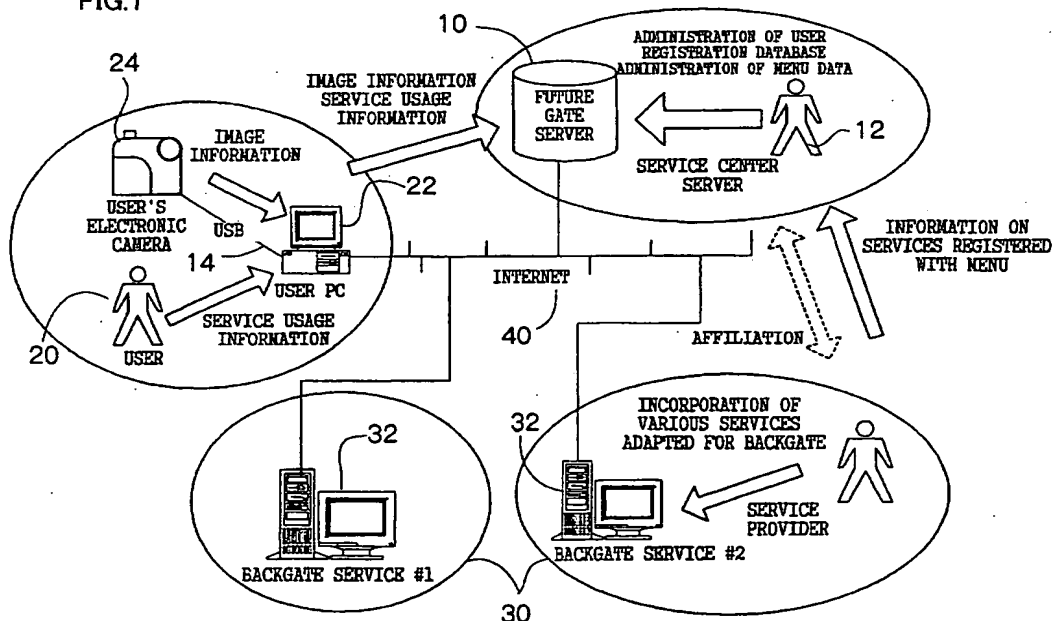
(30) Priority: **16.01.2001 JP 2001008068**

(71) Applicant: **Fuji Photo Film Co., Ltd.**
Minami-Ashigara-shi, Kanagawa (JP)

(54) Client/server system and button updating method for client application

(57) The client/server system enables functions of GUI buttons provided on a client application (50) to be easily changed. The client application (50) connects to a menu server (10) residing on a network (40) to download information on menu buttons (51). Button information includes button IDs unique to the menu buttons (51), parameter values specifying functions of the menu but-

tons (51), and information on images of the menu buttons (51). The menu server (10) uses user personal information to distribute menu button information suitable to each user (20). The client application (50) reads button IDs from menu information obtained from the menu server (10) and compares them with button information saved in a storage area of a client computer (22) to avoid duplicate downloading of the same button information.

FIG.1

Description

[0001] The present invention relates to a client/server system on a wide-area information network such as the Internet, and in particular, to an updating technique of a client application which enables data such as images to be updated or downloaded.

[0002] Conventional client applications in a client/server system on the Internet have only predetermined communication functions. For example, a communication function that is activated when a corresponding button on the application is depressed (clicked) cannot be changed unless the application is upgraded.

[0003] However, even if the client application is provided with a communication function of connecting to particular URLs (Uniform Resource Locators), the fixed URLs do not allow contents services to be freely constructed or allow the client application to connect directly to newly started contents services.

[0004] The present invention is provided in view of these problems, and it is an object of the preferred embodiments to provide a client/server system that allows the functions of GUI buttons provided on a client application to be easily changed.

[0005] To attain the above object, a first aspect of the present invention is directed to a client/server system comprising a plurality of computers connected to a network, wherein: a server on the network possesses button information which is data on menu buttons operating in connection with a client application introduced into a client computer, and the server has a function of transmitting the button information to the client computer; and the client application comprises a program which causes the client computer to provide a function of communicating with the server to obtain the button information from the server, a function of displaying menu buttons on a display in combination with a GUI screen of the client application according to the button information obtained, and a function of performing operations defined for the displayed menu buttons.

[0006] According to this aspect of the present invention, the client application connects to the server (menu server) present on the network to download the button information. The button information can be composed of parameter values required for the client application to execute various processes as well as button images. The operation of the client application performed when a button is pressed can be specified changing the values of parameters included in the button information. For example, for an image transfer button, the upper limit of the number of images that can be transferred can be set at a different value for each service connected using the corresponding button.

[0007] The server can provide information on menu buttons to the client in response to a request from the client application, and the client can download the information to display the various menu buttons. This enables the functions of GUI buttons provided on the client

application to be easily changed, thereby allowing users to change the functions to new ones. In particular, for a function of connecting to services provided on the network, service providers can allow the types or contents of the menu buttons to be easily changed according to slight changes in services.

[0008] Preferably, the button information includes button IDs as unique identification codes defined for the menu buttons, condition flags used to determine whether the menu buttons are enabled or disabled, action types which are condition flags used to determine operation of the menu buttons, and information used to identify images of the menu buttons.

[0009] Preferably, an effective start date and time and an effective end date and time are set as parameters for the button information; and the client application provides a function of displaying the menu buttons only during this period. Thus, by setting the effective start date and time and the effective end date and time as parameters for the button information, and displaying menu buttons only when the time of the client application or computer is within this period, the present invention is applicable to connection buttons for services with specified terms.

[0010] Preferably, the GUI screen of the client application has an update button operated by a user to instruct the menu buttons to be updated; and when the update button is operated, the client application transmits an update request to the server, and in response to the update request, the server provides the button information to the client application.

[0011] Of course, the update request may be issued not only by using the update button but also automatically when the application is activated. Alternatively, the application may be programmed so that the update request is automatically issued at predetermined time intervals.

[0012] Preferably, the client application comprises an image viewer which causes the client computer to provide an image transmitting and receiving function and an image browsing function; and the menu buttons are image transmitting GUI buttons for which a destination of an image is set.

[0013] Preferably the server comprises: a database which stores personal information on users who activate the client application to access the server; and a distribution button determining device which determines contents of the menu buttons to be distributed to the users on the basis of the users' personal information; and the button information on the menu buttons determined by the distribution button determining device is delivered to the client application.

[0014] In this case, preferably, the personal information on the users is registered in the database using an online user registering function of the client application; upon registration, each user is provided with a user ID which is a unique identification code; and subsequent requests from the client application to the server are pro-

vided with the user ID so as to authenticate the user ID.

[0015] That is, upon receiving a menu update command from the client application, the server can use the already collected personal information for each user to automatically create menus according to the users' tastes, and can transmit these menus to the client computers of the corresponding users. This method enables service providers to provide well-served marketing for the users.

[0016] Preferably, the server transmits list information on button IDs of new menu buttons to be incorporated, to the client application which has requested the current menu buttons to be updated; upon receiving the list information, the client application compares the button IDs described in the list information with the button IDs in the button information saved in a storage device of the client computer, and requests the server to obtain the button information on the button IDs described in the list information only if these button IDs are different from the button IDs in the button information; and the server transmits the button information on the requested button IDs to the client application.

[0017] The client application can read the IDs of the buttons written in the list information (menu information) and can compare these IDs with the button information saved in the storage area of the client computer, thereby preventing the duplicate downloading of the same buttons. This avoids unwanted communications.

[0018] A further aspect of the present invention is directed to a button updating method of a client application, comprising the steps of: constructing a client/server system by connecting client computers and a server together via a network; storing, in a menu button information database of the server, button information which is data on menu buttons operating in connection with a client application introduced into each of the client computers; activating the client application to communicate with the server to obtain button information therefrom; displaying the menu buttons on a display in combination with a GUI screen of the client application according to the button information obtained; and enabling operations defined for the displayed menu buttons.

[0019] Preferably, the button updating method further comprises the steps of: activating the client application to register personal information on users who access the server, in a user personal information database of the server; determining conditions for users to whom each menu button is distributed; checking the personal information on the users registered in the user personal information database against the conditions to determine menu buttons to be distributed to each user; and delivering button information on the determined menu buttons to the client application.

[0020] Preferably, the button information includes button IDs as unique identification codes defined for the menu buttons, condition flags used to determine whether the menu buttons are enabled or disabled, action types which are condition flags used to determine oper-

ation of the menu buttons, and information used to identify images of the menu buttons; the server transmits, to the client application which has requested the menu buttons to be updated, list information on button IDs of menu buttons to be incorporated; upon receiving the list information, the client application compares the button IDs described in the list information with the button IDs in the menu button information saved in a storage device of the client computer, and requests the server to obtain the button information on the button IDs described in the list information only if these button IDs are different from the button IDs in the button information; and the server transmits the button information on the requested button IDs to the client application.

[0021] The nature of this invention, as well as other objects and advantages thereof, will be explained in the following with reference to the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures and wherein:

Fig. 1 is a view showing an entire construction of a client/server system according to this embodiment; Fig. 2 is a view showing a construction of a system in which a user connects to a network using an image viewer;

Fig. 3 is a view showing a GUI screen of the image viewer;

Fig. 4 is a view used to describe an application of the system of this embodiment;

Fig. 5 is a flow chart showing a sequence used when a menu is updated;

Fig. 6 is a table showing contents of data contained in an event obtainment request;

Fig. 7 is a table showing the contents of data in menu information;

Fig. 8 is a view showing an example of menu information XML (service list XML);

Fig. 9 is a table showing the contents of data in button information;

Fig. 10 is a table showing the contents of data in button information;

Fig. 11 is a view showing an example of button information XML; and

Fig. 12 is a view used to describe a menu determining logic used by a menu server.

[0022] A preferred embodiment of a client/server system and a button updating method of a client application according to the present invention will be described in detail with reference to the accompanying drawings.

[0023] Fig. 1 is a view showing the entire construction of a system according to the embodiment of the present invention. This system is composed of a server (hereinafter referred to as an "operation administering server") 10 that administers the operation of the entire system, a computer (hereinafter referred to as a "user PC") 22 of a service user 20, and a server 32 (hereinafter referred to as a "service server") of a service provider 30

that provides printing and other services in response to requests from the user 20, the operation administering server 10, user PC 22, and service server 32 being connected to the Internet 40. The servers 10 and 32 each include a required database, and their communication specifications conform to, for example, the HTTP/1.0 and deal with the SSL (Secure Sockets Layer).

[0024] An administrator (hereinafter referred to as an "operation administrator") 12 that administers the operation of services provided by this system provides the operation administering server 10 and the service server 32 to operate services. Of course, the administrator need not possess the entire system by itself, but may implement the system by affiliating with the service provider 30. A plurality of operation administering servers 10 may be provided for each area (for example, each country) in which the services are provided.

[0025] The user 20 can communicate with the operation administering server 10 and the service server 32 using image browsing software with a communication function (hereinafter referred to as an "image viewer") which is incorporated in the user PC 22 or another communication terminal. The image viewer will be described later in detail, but comprises a function of uploading image data stored in a storage medium of an electronic camera 24 connected to the user PC 22, in the operation administering server 10, the service server 32, or other servers via the Internet 40 or downloading required data from any server.

[0026] The service provider 30 possesses a print process facility such as a printer (not shown), and prints photographs in response to an order from the user and delivers the resulting prints to the user, who has ordered the printing.

[0027] Fig. 2 is a view showing a construction of a system connected to the network via the image viewer. Fig. 3 is a view showing an example of a display screen (GUI screen) of the image viewer. In Fig. 2, a cradle 26 is connected to an electronic camera 24 via a communication interface such as an USB for communications, and can supply power to the electronic camera 24. The electronic camera 24 comprises an image pickup device which picks up an image of a subject and converts an optical image into an electric signal, a signal processing device that processes an image signal to convert it into a predetermined signal form, a recording device which records an image in a recording medium (a built-in memory or removable medium), and a communication device which transmits and receives information to and from other communication equipment. The electronic camera 24 may be provided with a microphone to record sound data therein.

[0028] The electronic camera 24 has a storage mode in which it functions as a storage device when connected to the user PC 22 or the like via the USB and a VC (Video Conference) mode in which it functions as a video camera used for television conferences to transmit video signals. The user can select either of the modes.

[0029] The user PC 22 automatically detects a connection of the electronic camera 24 via the communication device 28 such as the USB (using a plug & play function). When the electronic camera 24, set in the storage mode, is inserted into the cradle 26, the user PC 22 can automatically detect this connection and activate the image viewer 50 (see Fig. 3), application software that lists images from the electronic camera 24.

[0030] When the image viewer 50 is activated, the user PC 22 accesses the operation administering server 10 via the Internet 40 to obtain information such as a service menu for available services which is registered in the server 10. On the basis of the information obtained such as a service menu, as shown in Fig. 3, the window screen of the image viewer 50 displays various icons (also referred to as "menu buttons") 51.

[0031] In this regard, when a "user registration button" (not shown) is first depressed after the application of the image viewer 50 has been installed in the user PC 22, the user PC accesses a predetermined server (referred to as a "main server") to request a list of area names and the names of the corresponding local servers (area servers). The main server provides a list of area server names as required, and the user selects one of the local servers in the list by which he or she is controlled.

[0032] The user inputs personal information such as his or her name, address, zip code, e-mail address, type of electronic camera possessed, sex, telephone number, date of birth, occupation, and password to a user registration page on the selected local server. In response to the registration, a user ID is delivered to the user and is subsequently checked upon accesses to the server for authentication. The operation administering server 10 administers the personal information on the user 20 using the user ID. Thus, menu buttons 51 or advertisements customized for every user 20 can be distributed to him or her.

[0033] As shown in Fig. 3, the window of the image viewer 50 comprises a file list display section 53 that displays a folder structure, an image list display section 55 that lists thumbnail versions 54 of images stored in a folder designated by the user, and a menu display section 56 that lists menu buttons 51 for various services. The menu display section 56 may be partially used as an advertisement display section that displays banner advertisements or the like distributed by the operation administering server 10.

[0034] The menu buttons 51 are link buttons that allow connections to the server 32 of the service provider 30, described in Fig. 1, and include service selection buttons such as a print service button, a photo CD creation service button, a photograph publication button, a photo contest contribution button, an auction site transmission button, and cellular telephone display button.

[0035] The user 20 can transmit information such as images required to order a print service, receive a photo CD creation service or a service that publishes photographs on the Internet, or upload an image file for a con-

tribution to a photo contest or a sound file, by using a pointing device such as a mouse or an input device such as a keyboard (none of them are shown) to select one or more of the file names displayed in the file list display section 53 or one or more desired images from the list of thumbnail images 54 displayed in the image list display section 55, and then clicking the desired menu button 51.

[0036] Further, the image viewer 50 is provided with a menu update button 58. By clicking the menu update button 58 as required, the user can access the operation administering server 10 to obtain the latest menu button information.

[0037]—Now, a menu update function of the image viewer 50 constructed as described above will be described. First, the general flow of this function will be described by referring to Fig. 4. When the user clicks the menu update button 58 of the image viewer 50, the user PC 22 accesses the local server (corresponding to the operation administering server 10) having control over services, to obtain menu information, button information, and button images. The menu information is a collection of plural pieces of button information. The button information is composed of parameter values required for the image viewer 50 in which the buttons are incorporated, to execute various processes, as well as button images. At this time, the operation administering server 10 functions as a menu server that distributes information on the menu buttons 51 to the user PC 22 (the operation administering server 10 will be referred to as a "menu server" as required).

[0038] On the basis of the menu information, button information, and button images obtained from the menu server, the list of menu buttons 51 in the image viewer 50 of the user PC 22 is updated. Various parameters (for example, the URL of a destination) that determine the operation performed when each of the menu buttons 51 is depressed are included in the button information obtained from the menu server.

[0039] When the user depresses any of the menu buttons 51, the image viewer 50 activates a WWW browser 60, and the URL of the service site is delivered to the browser. Then, the service site specified by the URL is accessed to obtain a Web page from a server 62 for the service site.

[0040] Fig. 5 is a flow chart showing a sequence used when the menu is updated. In this figure, the left side shows a process executed by the client (user PC 22), while the right side shows a process executed by the menu server. When the user clicks the menu update button 58 on the image viewer 50 screen (step S110), an event obtainment request is transmitted by the user PC 22 (step S112). The "event" as used herein refers to a "special plan" or the "latest news", that is, information presented to the user by the operation administrator. Fig. 6 shows the contents of data contained in the event obtainment request.

[0041] In Fig. 5, the menu server receives the event

obtainment request (step S114) and authenticates transmitted user ID and password (step S116). This authentication process is executed by checking the user ID and the password against the user personal information registered in a database 13. If the authentication fails in step S116, the menu server transmits a status code to the client, indicating that the "authentication has failed". When the client receives the authentication failure notification (step S117), the process jumps to processing in step S124.

[0042] If the authentication succeeds in step S116, the process proceeds to step S118 to create event XML. In this case, an event that meets the user's conditions is extracted utilizing event information stored in a database 14, and described in an XML (Extensible Markup Language) form.

[0043] The thus created event XML is transmitted to the client (step S120). When the client receives the result of the event determination (event XML) transmitted from the menu server (step S122), information on the event is displayed on the image viewer 50.

[0044] Subsequently, the user PC 22 transmits a menu information obtainment request to the menu server (step S124). Upon receiving the menu information obtainment request (step S126), the menu server authenticates transmitted user ID and password (step S128). This authentication process is executed by checking the user ID and the password against the user personal information registered in the database 13. If the authentication fails in step S128, the menu server transmits a status code to the client, indicating that the "authentication has failed". When the client receives the authentication failure notification (step S129), the menu update process sequence is ended (step S130).

[0045] If the authentication succeeds in step S128, the process proceeds to step S132. At step S132, a menu update count is incremented by one to update the count value. The count value of the number of menu updates is stored in the database 13 as user personal information.

[0046] Then, the process proceeds to step S134 to create a service list XML that meets the user's conditions. In this case, with reference to a menu determination condition database 15 and the user personal information database 13, menu buttons ID that meet the conditions are determined on the basis of the personal information on the user who has issued the menu obtainment request as well as menu determination conditions. The button IDs are identification codes uniquely (without any duplicates) defined for the menu buttons 51. Further, messages and other message information displayed on the image viewer 50 during a menu distribution process are extracted from a message information database 16. The list of button IDs and message information thus determined are described in the XML form. At this time, the display priorities of the menu buttons 51 are properly determined according to the user's conditions. A determination logic for the service list dis-

played to the user will be described in detail with reference to Fig. 12.

[0047] The service list XML created in step S134 in Fig. 5 is transmitted to the user PC 22 of the client as the result of the menu information retrieval (step S136). Fig. 7 shows the contents of data transmitted from the menu server to the image viewer 50 as menu information. Fig. 8 shows an example of menu information XML (service list XML).

[0048] In Fig. 5, upon receiving the service list XML (step S138), the user PC 22 analyzes the contents of the description (step S140) to determine button IDs to be obtained (step S142). That is, the image viewer 50, the client application, reads the button IDs described in the service list XML, and compares them with the button IDs saved in a service area of the user PC 22 to determine whether or not any of the button IDs in the service list has been updated. Then, the image viewer 50 determines that button information be obtained for only the added or changed buttons. This prevents the duplicate downloading of the same button information as that already retained in the user PC 22, thereby avoiding unwanted communications.

[0049] When the button IDs to be obtained are determined in step S142, it is determined whether or not one or more buttons are to be obtained (step S144). If no button is to be obtained (the result of the determination is negative), the process jumps to step S168 to display button images in the menu display section 56 of the image viewer 50 on the basis of the button information and images retained in the user PC 22 (step S168), thereby a load completion message is displayed (step S170) to end this sequence (step S172).

[0050] If it is determined in step S144 that at least one button be obtained (the result of the determination is affirmative), the process proceeds to step S146 to display a message on the screen of the image viewer 50, indicating that loading is being carried out. Then, a button information obtainment request is transmitted (step S148). The button information obtainment request contains data on session IDs and button IDs.

[0051] Upon receiving the button information obtainment request, the menu server retrieves button information on the button IDs in the request from a menu button information database 17 to create button information XML on the basis of the result of the retrieval (step S150). The thus created button information XML is transmitted to the client (step S152). Figs. 9 and 10 show the contents of data transmitted from the menu server to the image viewer 50 as button information. Fig. 11 shows an example of button information XML.

[0052] At step S152 in Fig. 5, if the client receives the button information XML transmitted by the menu server (step S154), the client application analyzes this information (step S156). Then, the client obtains the URL of the destination of button image files which is contained in the button information, and transmits a button image obtainment request to the menu server (step S158).

[0053] The menu server retrieves button images relating to the request, from a button image database 18 (step S160), and outputs image files with the button images (step S162).

5 [0054] The client application receives the button images transmitted by the menu server (step S164), and subsequently determines whether or not there are any button IDs for which no button information has been obtained (step S166). If there are any button IDs for which no button information has been obtained, the process returns to step S148 to repeat the above described process (steps S148 to S166) for all the button IDs for which button information has not been obtained to obtain required button information and images.

10 [0055] At step S166, if it is determined that button information has been obtained for all the button IDs, the process proceeds to step S168 to display the button images in the menu display section 56 of the image viewer 50. Then, after a load completion message is displayed (step S170), this sequence is completed (step S172). Thus, the menu display section 56 of the image viewer 50 displays the latest menu buttons 51 optimized for the user. In this example, the URL of the location in which the image files are stored is used as information used to identify the button images, but data on the images themselves may be contained in the button information, which may then be provided to the client by the server.

15 [0056] The operation of each menu button 51 is determined by parameters in the button information, and the image viewer 50 determines the operation of the menu button according to the parameters in the button information. For example, on the basis of the result of the determination of whether or not the electronic camera is connected to a USB cable, in what mode the electronic camera is, or whether or not the image viewer is selecting images, the electronic viewer 50 determines button images to be displayed (enable or disable the buttons) and strings to be popped up when the pointer is placed on the corresponding buttons.

20 [0057] Furthermore, the image viewer 50 determines the following on the basis of the parameters in the button information: #1. The image viewer 50 determines URLs and their options that are delivered to the browser when the corresponding menu buttons 51 are depressed. #2. The image viewer 50 determines whether or not to transmit images when the corresponding menu buttons 51 are depressed. #3. The image viewer 50 determines the numbers of images, the sizes of images (longitudinal and lateral pixels), file sizes (capacities), and file types or the like that can be transmitted when the corresponding menu buttons are depressed. #4. The image viewer 50, when activated, determines whether or not the terms of validity of the buttons, specified by appear and disappear dates, have expired.

25 [0058] Next, logic that determines a service list presented to the user on the basis of the user personal information will be explained with reference to Fig. 12.

[0059] As shown in this drawing, a menu server 70

(operation administering server 10) has the menu buttons information database 17, the menu determination condition database 15, and the user personal information database 13. The administrator 72 (corresponding to the operation administrator 12) of the menu server registers and administers menu button information and menu determination conditions. The menu button information database 17 stores button information for each button ID, indicating the contents of the operation of the corresponding button.

[0060] The menu determination condition database 15 stores data specifying conditions for users to whom menus are to be transmitted. The conditions for the users to whom menus are to be transmitted are described for each button ID; a button ID 1 indicates no conditions, a button ID 2 indicates a woman, a button ID 3 indicates a student or a user of the digital camera "FinePix40i", a button ID 4 indicates a man of thirty or older, and a button ID 5 indicates no conditions.

[0061] Users (A and B) register their personal information online, which is stored in the user personal information database 13. For example, the user A's personal information includes "male", "40 years old", "civil servant", and "FinePix40i user". The user B's personal information includes "female", "21 years old", "student", and "FinePix40i user".

[0062] In this case, when the user A requests a menu from the menu server 70, the menu server 70 transmits {button ID 1, button ID 3, button ID 4, button ID 5 ...} to the user A as the result of a check against the menu determination conditions.

[0063] Further, when the user B requests a menu from the menu server 70, the menu server 70 transmits {button ID 1, button ID 2, button ID 3, button ID 5 ...} to the user B as the result of a check against the menu determination conditions. The subsequent process is as described for step S138 and the subsequent steps in Fig. 5.

[0064] The usage (history) of the menu server 70 or service provider 30 by the user is recorded in the operation administering server 10 as personal information. When a user ID is received from the user, information to be distributed is selected on the basis of the history for the user. Further, frequently used information from the service provider 30 may be given a higher priority, display priorities may be mutually changed, or frequently used menu buttons 51 may be displayed at such locations that these buttons are more easily selected by the user.

[0065] In the above description, the button update process is executed in response to the depression of the menu update button 58, but it may be automatically executed when the image viewer 50 is activated or periodically while the computer is online. Alternatively, the user may be asked if he or she allows the update process to be executed, when he or she depresses any of the menu buttons 51 to access the server. Alternately, if the image viewer 50 or the menu server 70 administers

the history of updates and no updates have been carried out for a long time, the user may be warned about the need for an update process.

[0066] In the above embodiment, the image viewer 50 that downloads and uploads images has been described, but the application of the present invention is not limited to it. The present invention is applicable to computer programs handling various electronic data such as still images, animated images, sounds, music, and books (documents).

[0067] As described above, the server provides information on menu buttons in response to a request from the client application, and the client downloads this information to display the various menu buttons. Consequently, the functions and display forms (including button images and arrangements) of the GUI buttons incorporated in the client application can be easily changed. Further, by using user's personal information menu buttons are distributed to the user according to his or her tastes, thereby providing well-served marketing for him or her.

[0068] Furthermore, a list of relevant button IDs is transmitted to the client application which has requested the menu buttons to be updated, and this information is compared with the button information saved in the storage area of the client computer. This prevents the duplicate downloading of the same button information to avoid unwanted communications.

[0069] It should be understood, however, that there is no intention to limit the invention to the specific forms disclosed, but on the contrary, the invention is to cover all modifications, alternate constructions and equivalents falling within the spirit and scope of the invention as expressed in the appended claims.

Claims

1. A client/server system comprising a plurality of computers (10, 22, 32) connected to a network (40), wherein:

a server (10) on the network (40) possesses button information which is data on menu buttons (51) operating in connection with a client application (50) introduced into a client computer (22), and the server (10) has a function of transmitting the button information to the client computer (22); and

the client application (50) comprises a program which causes the client computer (22) to provide a function of communicating with the server (10) to obtain the button information from the server (10), a function of displaying menu buttons (51) on a display in combination with a GUI screen of the client application (50) according to the button information obtained, and a function of performing operations defined for the

displayed menu buttons (51).

2. The client/server system according to claim 1, wherein the button information includes button IDs as unique identification codes defined for the menu buttons (51), condition flags used to determine whether the menu buttons (51) are enabled or disabled, action types which are condition flags used to determine operation of the menu buttons (51), and information used to identify images of the menu buttons (51).

3. The client/server system according to claim 1 or 2, wherein:

an effective start date and time and an effective end date and time are set as parameters for the button information; and
the client application (50) provides a function of displaying the menu buttons (51) only during this period.

4. The client/server system according to any one of claims 1 to 3, wherein:

the GUI screen of the client application (50) has an update button (58) operated by a user (20) to instruct the menu buttons (51) to be updated; and
when the update button (58) is operated, the client application (50) transmits an update request to the server (10), and in response to the update request, the server (10) provides the button information to the client application (50).

5. The client/server system according to any one of claims 1 to 4, wherein:

the client application (50) comprises an image viewer which causes the client computer (22) to provide an image transmitting and receiving function and an image browsing function; and the menu buttons (51) are image transmitting GUI buttons for which a destination of an image is set.

6. The client/server system according to any one of claims 1 to 5, wherein:

the server (10) comprises:

a database (13) which stores personal information on users (20) who activate the client application (50) to access the server (10); and
a distribution button determining device which determines contents of the menu buttons (51) to be distributed to the users

(50) on the basis of the users' personal information; and

the button information on the menu buttons (51) determined by the distribution button determining device is delivered to the client application (50).

7. The client/server system according to claim 6, wherein:

the personal information on the users (20) is registered in the database (13) using an online user registering function of the client application (50);
upon registration, each user (20) is provided with a user ID which is a unique identification code; and
subsequent requests from the client application (50) to the server (10) are provided with the user ID so as to authenticate the user ID.

8. The client/server system according to claim 2, wherein:

the server (10) transmits list information on button IDs of new menu buttons (51) to be incorporated, to the client application (50) which has requested the current menu buttons (51) to be updated;
upon receiving the list information, the client application (50) compares the button IDs described in the list information with the button IDs in the button information saved in a storage device of the client computer (22), and requests the server (10) to obtain the button information on the button IDs described in the list information only if these button IDs are different from the button IDs in the button information; and
the server (10) transmits the button information on the requested button IDs to the client application (50).

9. A button updating method of a client application (50), comprising the steps of:

constructing a client/server system by connecting client computers (22) and a server (10) together via a network (40);
storing, in a menu button information database (17) of the server (10), button information which is data on menu buttons (51) operating in connection with a client application (50) introduced into each of the client computers (22);
activating the client application (50) to communicate with the server (10) to obtain button information therefrom;
displaying the menu buttons (51) on a display

in combination with a GUI screen of the client application (50) according to the button information obtained; and
enabling operations defined for the displayed menu buttons (51).

5

10. The button updating method according to claim 9, further comprising the steps of:

activating the client application (50) to register
personal information on users (20) who access
the server (10), in a user personal information
database (13) of the server (10);
determining conditions for users (20) to whom
each menu button (51) is distributed;
checking the personal information on the users
(20) registered in the user personal information
database against the conditions to determine
menu buttons (51) to be distributed to each user
(20); and
delivering button information on the determined
menu buttons (51) to the client application (50).

10

15

20

11. The button updating method according to claim 9 or 10, wherein:

25

the button information includes button IDs as
unique identification codes defined for the
menu buttons (51), condition flags used to de-
termine whether the menu buttons (51) are en-
abled or disabled, action types which are con-
dition flags used to determine operation of the
menu buttons (51), and information used to
identify images of the menu buttons (51);
the server (10) transmits, to the client applica-
tion (50) which has requested the menu buttons
(51) to be updated, list information on button
IDs of menu buttons (51) to be incorporated;
upon receiving the list information, the client
application (50) compares the button IDs de-
scribed in the list information with the button IDs
in the menu button information saved in a stor-
age device of the client computer (22), and re-
quests the server (10) to obtain the button in-
formation on the button IDs described in the list
information only if these button IDs are different
from the button IDs in the button information;
and
the server (10) transmits the button information
on the requested button IDs to the client appli-
cation (50).

30

35

40

45

50

55

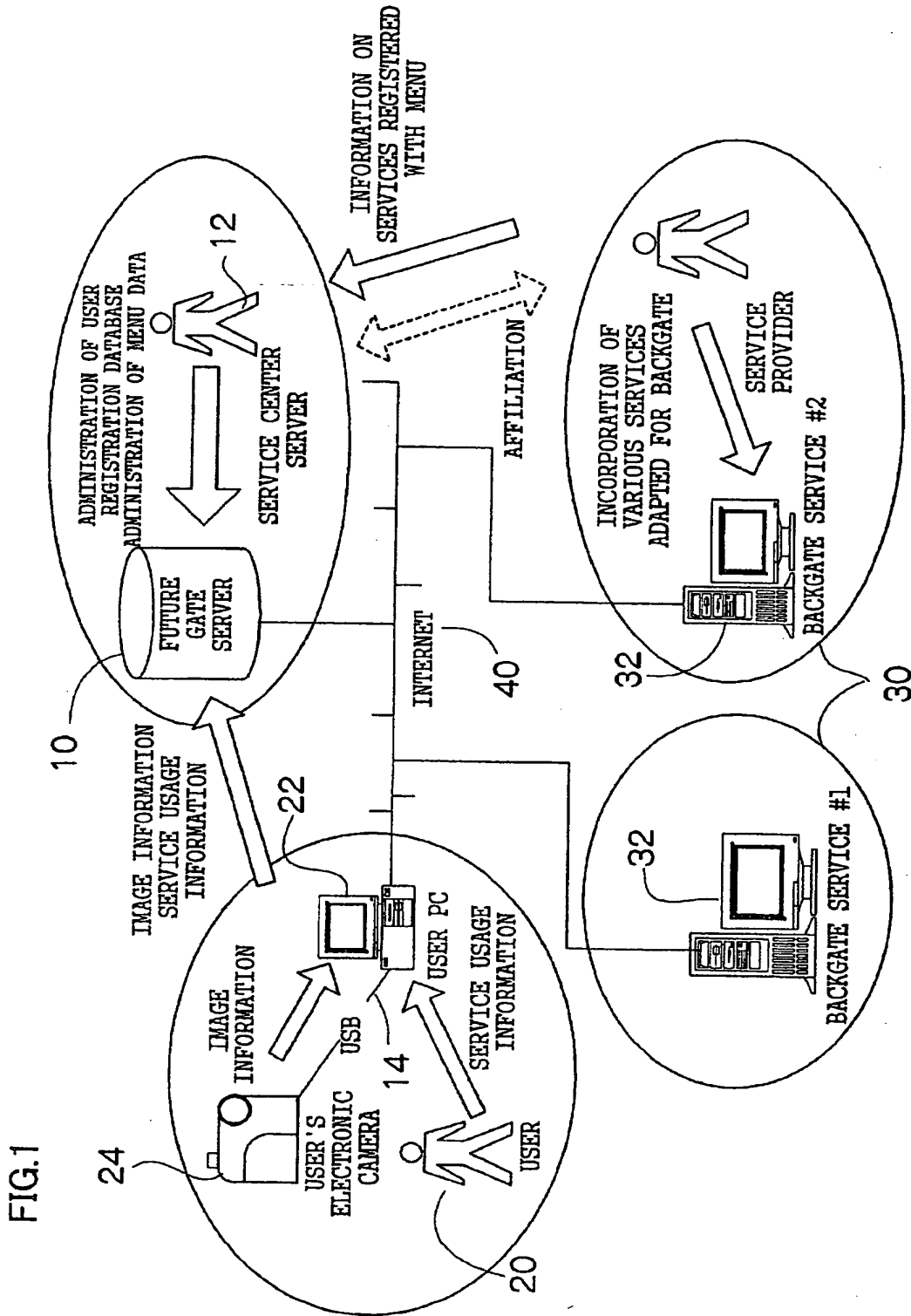


FIG. 2

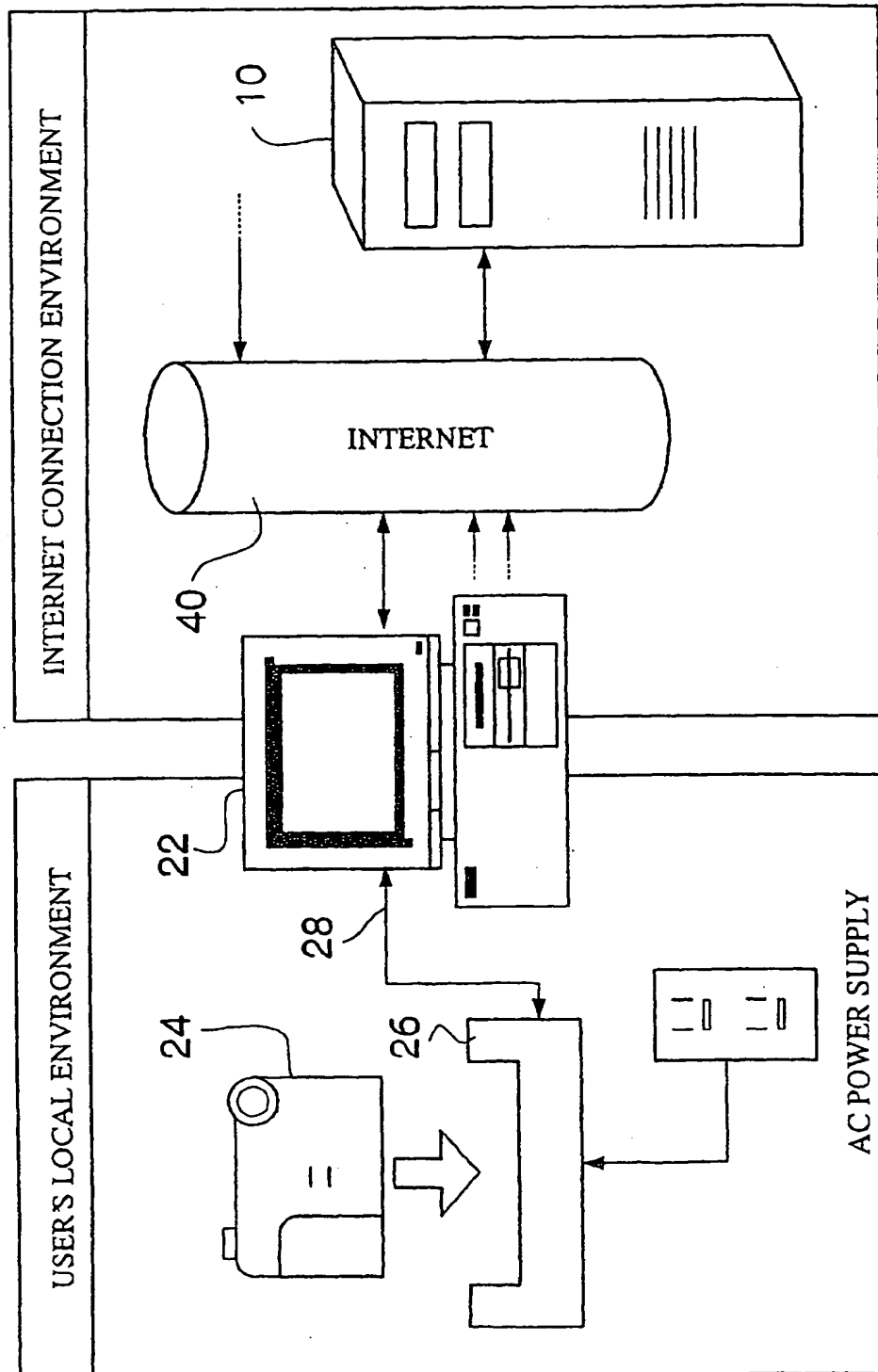


FIG.3

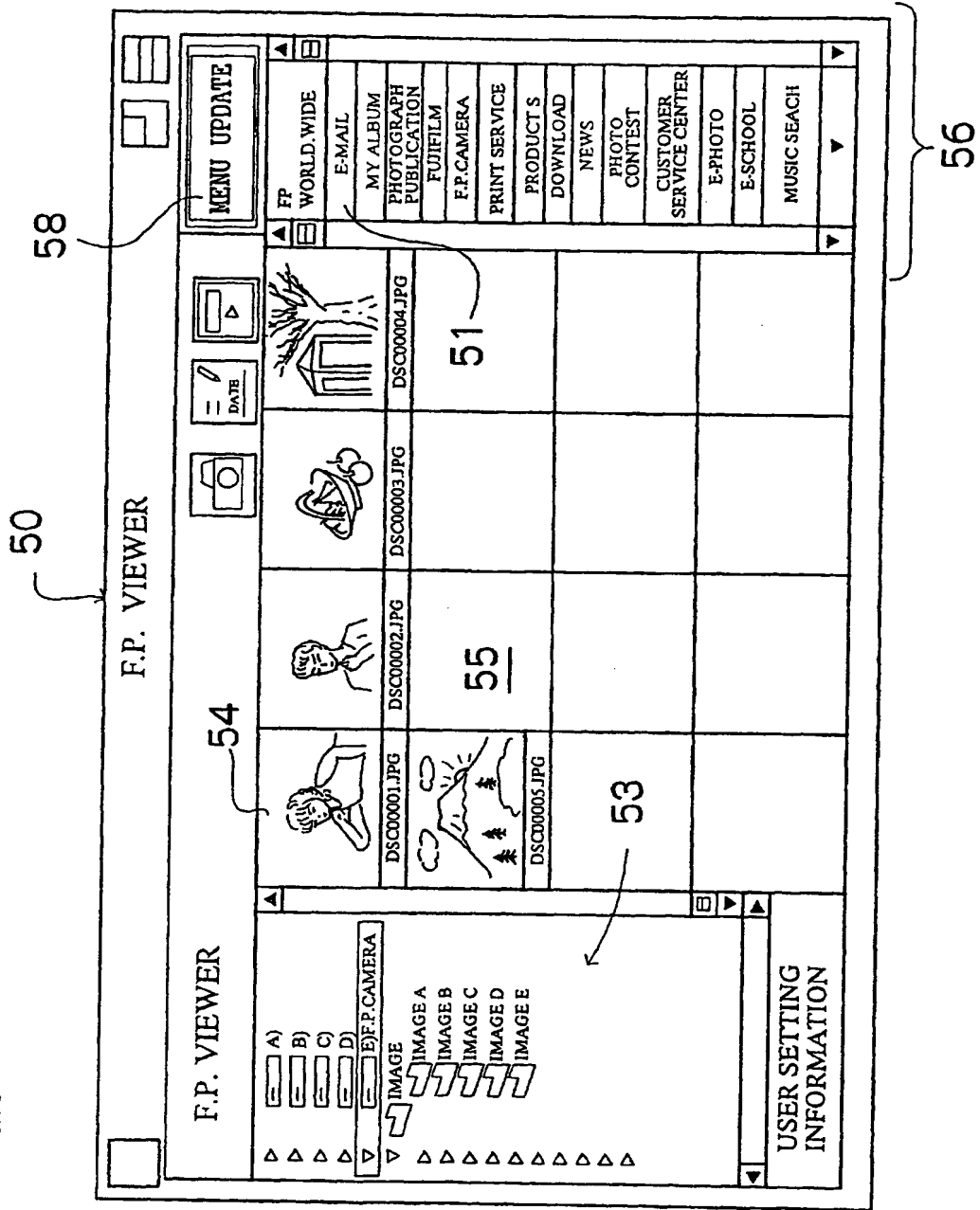


FIG.4

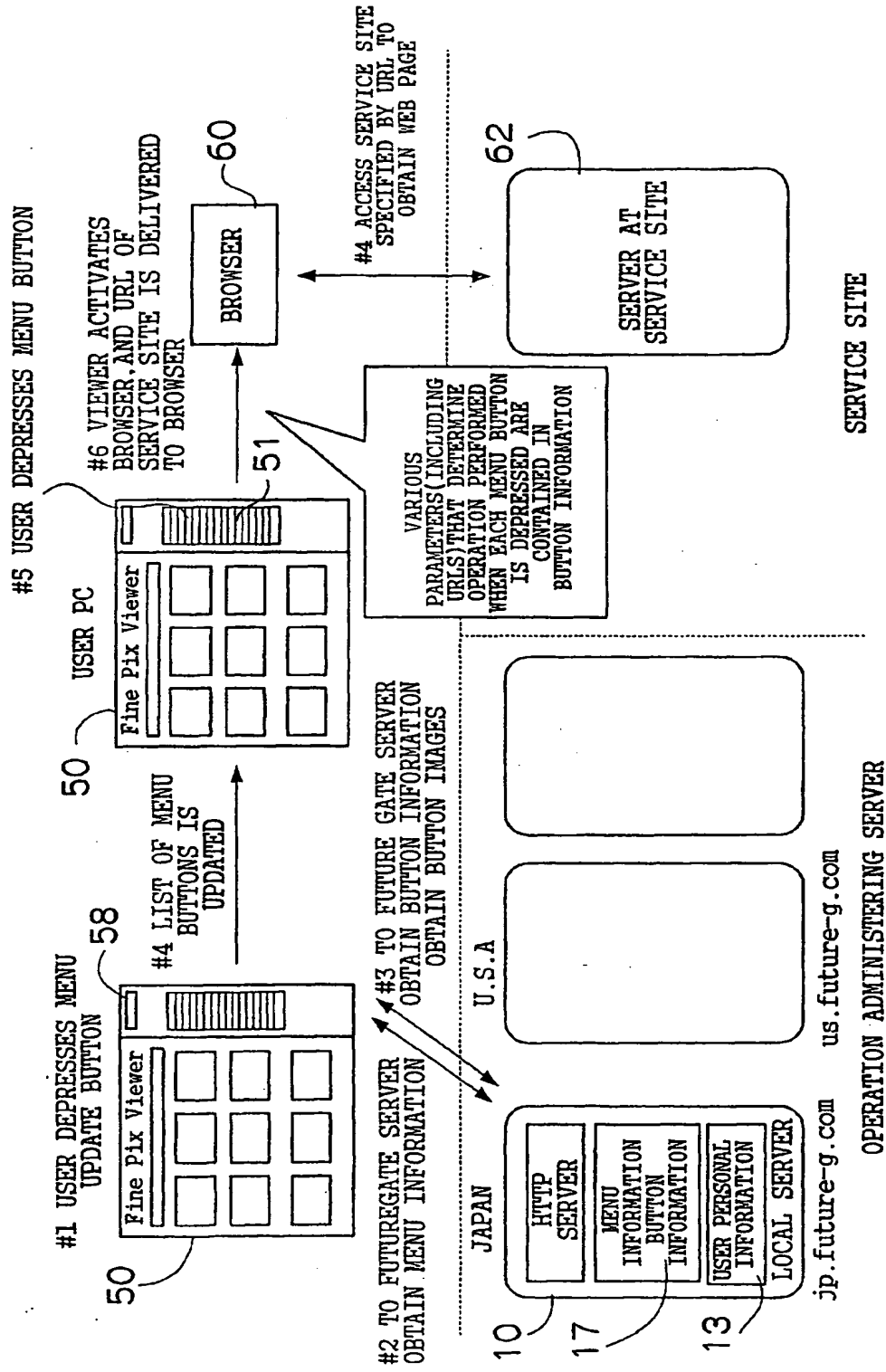


FIG.5

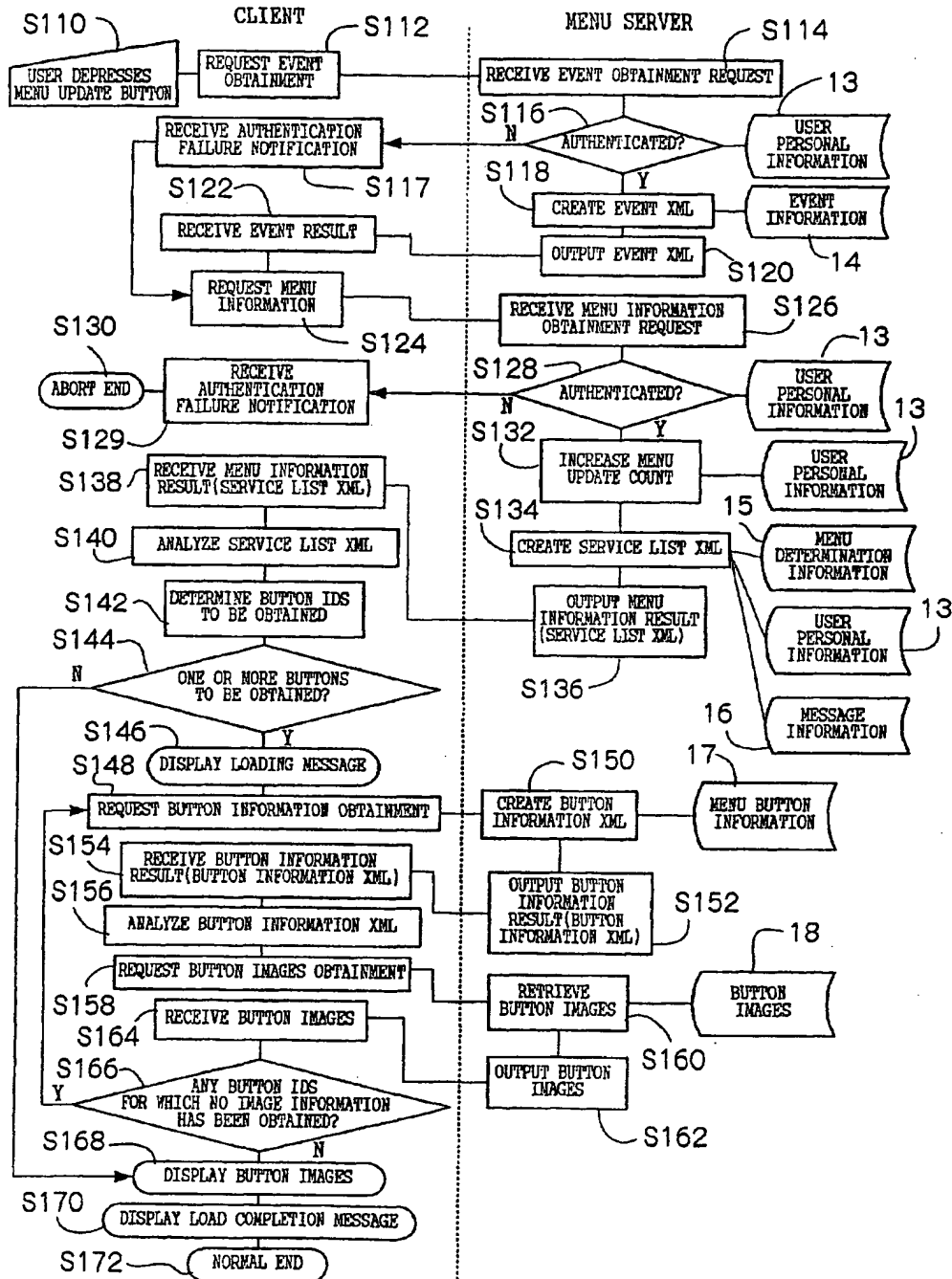


FIG.6

EVENT OBTAINMENT REQUEST

| REQUEST | | REMARKS |
|--------------------|------------------------------|---|
| ACCESS RESOURCE | xx_future-g.com/viewer/event | |
| PROPERTY | PROPERTY | NAME |
| | USER ID | USER ID |
| | USER PW | PASSWORD |
| | EXE TIMES | NUMBER OF VIEWER ACTIVATION OPERATIONS |
| | TOTAL MINUTES | TOTAL TIME FOR WHICH VIEWER HAS BEEN USED |
| | LATEST INTERVAL | NUMBER OF DAYS THAT HAVE PASSED SINCE LAST USE OF VIEWER |
| | (TEST FLAG) | (FLAG FOR TESTS) |
| | | PROPERTY THAT CAN BE ADDED USING VIEWER OPTION TOOL (TESTFLAG=1) |

FIG.7

DATA TRANSMITTED FROM MENU SERVER TO CLIENT AS MENU INFORMATION

| ELEMENT (PARAMETER NAME) | DATA FORMAT | DISCRIPTION |
|-----------------------------|---|--|
| session-id | 64 OR LESS LATERALLY HALF CHARACTERS | STRING ISSUED BY SERVER |
| tool-tip | 255BYTES | DESCRIPTION TEXT FOR MENU |
| loading-msg-id | INTEGER(INT 4BYTES) | 0-4294967295 0-1000000000 FOR EACH TIME DISPLAY 1000000001-FOR SINGLE DISPLAY |
| loading-msg-sentence | 255BYTES | MESSAGE TEXT |
| complete-msg-id | INTEGER(INT 4BYTES) | 0-4294967295 0-1000000000 FOR EACH TIME DISPLAY 1000000001-FOR SINGLE DISPLAY |
| complete-msg-sentence | 255BYTES | MESSAGE TEXT |
| complete-msg-url | 255BYTES | URL OF WWW PAGE OPENED WHEN LOADING IS COMPLETED |
| button-count | INTEGER(INT 4BYTES) | NUMBER OF BUTTONS |
| button-id | INTEGER(INT 4BYTES) | BUTTON IDS(ONE OR MORE) 0-4294967295 |

FIG.8

EXAMPLE OF TRANSMITTED MENU INFORMATION XML

```

<?xml version="1.0"encoding="UTF-8">
<!DOCTYPE response SYSTEM"servicelist.dtd">

<response>

  <session-id>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</session-id>

  <service-list>

    <tool-tip>WELLCOME .TO FinePix INTERNET SERVICE!</tool-tip>

    <loading-msg-id>l2</loading-msg-id>

    <loading-msg-sentence>DURING LOAD</loading-msg-sentence>

    <complete-msg-id>l3</complete-msg-id>

    <complete-msg-sentence>LOAD COMPLETE</complete-msg-sentence>

    <complete-msg-url></complete-msg-url>

    <button-count>5</button-count>

    <button-id>l</button-id>

    <button-id>45</button-id>

    <button-id>56</button-id>

    <button-id>22</button-id>

    <button-id>4</button-id>

  </service-list>

</response>

```

FIG.9

CONTENTS OF DATA TRANSMITTED FROM MENU SERVER TO CLIENT
AS BUTTON INFORMATION

| ELEMENT (PARAMETER NAME) | DATA FORMAT | DESCRIPTION |
|-----------------------------|--|--|
| button-id | INTEGER(INT 4BYTES) | BUTTON ID 0-4294967295 |
| service-id | INTEGER(INT 4BYTES) | SERVICE ID 0-4294967295 |
| caption | 255BYTES | BUTTON NAME TEXT*RESERVATION TAG |
| view-category | | DISPLAY CATEGORY*RESERVATION TAG |
| service-name | 255BYTES | SERVICE NAME |
| condition-flag | 0x0000=OPERATE REGARDLESS OF CONDITIONS 0x0001=ENABLED ONLY DURING CONNECTION TO ACTUAL MACHINE(STORAGE MODE) 0x0002=ENABLED ONLY DURING CONNECTION TO ACTUAL MACHINE(VG MODE) 0x0100=ENABLED ONLY DURING SELECTION OF IMAGES 0x8000=ALWAYS DISABLED | CONDITION FLAG USED TO DETERMINE WHETHER BUTTON IS TO BE ENABLED OR DISABLED (ONE OR MORE LOGICAL ORS) EXAMPLE) 0x0103=CONNECTION TO ACTUAL MACHINE AND IMAGE SELECTION REQUIRED |
| action-type | 0x0001:ONLY SERVICE ACCESS COUNT 0x0010:ONLY IMAGE TRANSFER USING GFNAP2 0x0100:ONLY USER INFORMATION TRANSMISSION | CONDITION FLAG USED TO DETERMINE OPERATION WHEN BUTTON IS DEPRESSED (ONE OR MORE LOGICAL ORS) |
| upload-img-min | INTEGER(INT 4BYTES) | MINIMUM NUMBER OF TRANSFERRED IMAGES |
| upload-img-max | INTEGER(INT 4BYTES) | MAXIMUM NUMBER OF TRANSFERRED IMAGES |
| upload-img-type | 0X0001:STILL IMAGE FILE 0X0010:ANIMATED IMAGE FILE 0x0100:SOUND FILE | CONDITION FLAG USED TO DETERMINE TYPE OF FILE THAT CAN BE UPLOADED(LOGICAL OR AVAILABLE) |
| link-address | 255BYTES | URL OF WWW PAGE FOR CONNECTION SERVICE |
| link-option | 255BYTES | ARGUMENT FOR LINK ADDRESS |
| enable-tool-tip | 255BYTES | BALLOON HELP TEXT FOR ENABLED BUTTON |

FIG.10

| | | |
|------------------------------|------------------------------------|--|
| disable-tool-tip | 255BYTES | BALLOON HELP TEXT FOR ENABLED BUTTON |
| enable-button | 255BYTES | URL OF IMAGE FILE |
| disable-button | 255BYTES | URL OF IMAGE FILE |
| img-x-min (RESERVED) | INTEGER (INT 4BYTES) | MINIMUM NUMBER OF PIXELS (LATERAL) |
| img-y-min (RESERVED) | INTEGER (INT 4BYTES) | MINIMUM NUMBER OF PIXELS (LONGITUDINAL) |
| img-x-max (RESERVED) | INTEGER (INT 4BYTES) | MAXIMUM NUMBER OF PIXELS (LATERAL) |
| img-y-max (RESERVED) | INTEGER (INT 4BYTES) | MAXIMUM NUMBER OF PIXELS (LONGITUDINAL) |
| img-size-max (RESERVED) | INTEGER (INT 4BYTES), UNIT: KBYTES | MAXIMUM FILE SIZE |
| img-format (RESERVED) | EXTENSION DESCRIBED USING COLONS | TRANSMITTABLE IMAGE FORMAT |
| appear-date (RESERVED) | yyyymmdd | DISPLAY START DATE AND TIME |
| disappear-date (RESERVED) | yyyymmdd | DISPLAY END DATE AND TIME |

FIG.11

EXAMPLE OF TRANSMITTED BUTTON INFORMATION XML

```

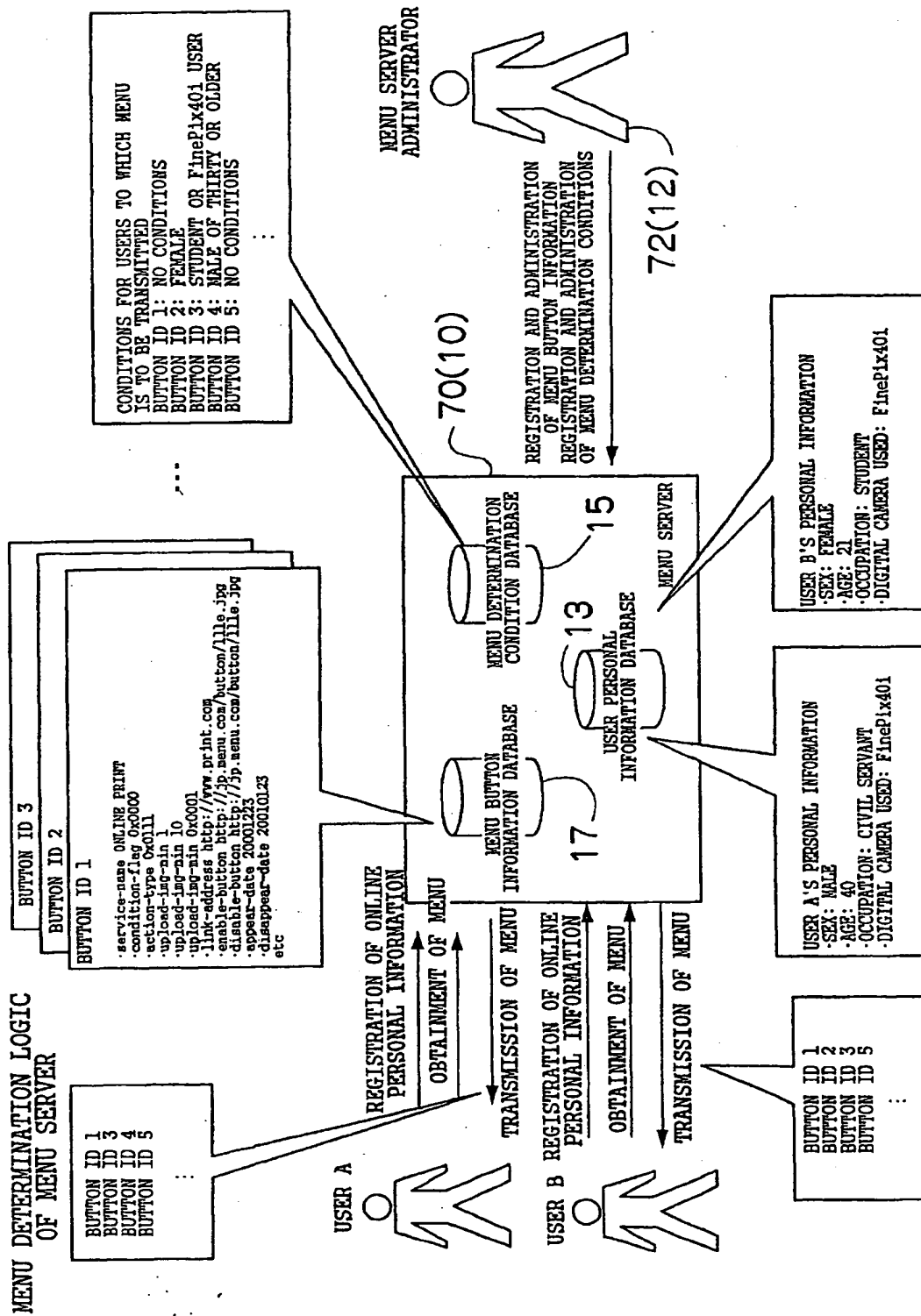
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE response SYSTEM "button.dtd">
<response>

    <button-id>1</button-id>
    <service-id>1</service-id>
    <caption>FD1 SERVICE</caption>
    <view-category>3</view-category>
    <service-name>fdi</service-name>
    <condition-flag>0x0100</condition-flag>
    <action-type>0x0001</action-type>
    <upload-img-min>1</upload-img-min>
    <upload-img-max>3</upload-img-max>
    <upload-img-type>0x0001</upload-img-type>
    <link-address>http://www.xxxxxxx.co.jp</link-address>
    <link-option></link-option>
    <enabled-tool-tip>UP TO THREE IMAGES CAN BE SELECTED</enabled-tool-tip>
    <disabled-tool-tip>SELECT IMAGES</disabled-tool-tip>
    <enabled-button>http://jp.future-g.com/viewer/button/xxxxxxx.jpg</enabled-button>
    <disabled-button>http://jp.future-g.com/viewer/button/xxxxxxx.jpg</disabled-button>
    <img-x-min>320</img-x-min>
    <img-y-min>240</img-y-min>
    <img-x-max>640</img-x-max>
    <img-y-max>480</img-y-max>
    <img-size-max>100</img-size-max>
    <img-format>jpg:gif:bmp</img-format>
    <appear-date>20010401</appear-date>
    <disappear-date>20010433</disappear-date>

</response>

```

FIG.12



THIS PAGE BLANK (USP...)